

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listing, of claims in the application:

1. (Currently Amended) An abstraction layer for interfacing a computer to a telephony radio, comprising:  
a set of application programming interfaces (APIs) for abstracting out multiple radio technologies without knowledge of the telephony radio or cellular network, wherein the set of APIs correspond to call control functions, wherein the abstraction layer comprises a proxy layer and a driver layer, wherein when the proxy layer receives a call at a first interface to one of the set of APIs is called, the proxy layer transforms the API call to a command understood by the driver layer and sends the command to the driver layer at a second interface, and wherein the driver layer receives the command at the second interface and ~~abstraction layer~~ determines at least one standard telephony radio command corresponding to the called API and sends the telephony radio command to the telephony radio at a third interface, ~~and wherein the abstraction layer comprises a proxy layer and a driver layer.~~
2. (Original) The abstraction layer of claim 1 wherein the telephony radio is one of a plurality of telephony radios which operates based on the standard telephony radio commands.
3. (Original) The abstraction layer of claim 1 wherein the set of APIs further correspond to short messaging system functions.
4. (Original) The abstraction layer of claim 3 wherein the set of APIs further correspond to network service functions.
5. (Original) The abstraction layer of claim 4 wherein the set of APIs further correspond to data connection functions.

6. (Original) The abstraction layer of claim 5 wherein the set of APIs further correspond to interface functions.

7. (Currently Amended) A radio interface layer of a telephone for facilitating communications between an application program module and a radio, comprising:

a proxy layer for communicating with the application program at a first interface and a driver layer at a second interface, wherein the proxy layer provides an API on the first interface for receiving application program calls ~~an API~~ to perform a particular function and wherein the proxy layer transforms the API calls to an input/output control (IOCTL) code and sends the IOCTL code to the driver layer at the second interface; and

wherein the driver layer communicates with the proxy layer at the second interface and the radio at a third interface, the driver layer receiving an IOCTL code at the second interface and transforming the IOCTL code into a command understood by the radio to perform the particular function and sending the radio command at the third interface.

8. (Original) The radio interface layer of claim 7 wherein the driver layer further receives communications from the radio indicating that the particular function has been performed and wherein the driver layer sends a success code to the proxy layer indicating that the particular function has been performed.

9. (Currently Amended) A method for processing commands in a telephone comprising a proxy layer, a driver layer, an application and a radio, the method comprising the steps of:

causing the application to call a radio interface layer (RIL) API in the proxy layer at a first interface, wherein the RIL API is associated with an action to be performed by the radio;

causing the proxy layer to translate the RIL API into IOCTL codes;

sending the IOCTL codes to the driver layer at a second interface;

translating the IOCTL codes to a command corresponding to the action, wherein the command will be understood by the radio; and

sending the command to the radio at a third interface.

10. (Original) The method of claim 9 wherein the command is an AT command.
11. (Original) The method of claim 9 wherein the command is one of a private API set defined by the radio manufacturer.
12. (Original) The method of claim 9 further comprising the step of generating in the RIL driver layer a unique ID associated with the RIL API.
13. (Original) The method of claim 12 further comprising the step of waiting for a response from the radio, and when received, calling back the calling application with the response and the unique ID returned from the call.
14. (Original) The method of claim 13 wherein the RIL driver matches the response from the radio with the unique ID and the RIL driver sends the response to the calling process via a callback function.
- 15 - 16. (Canceled)
17. (Currently Amended) A method of communicating between a module and a radio comprising:
  - (a) generating a RIL API call at one of a plurality of modules to perform a specific action;
  - (b) sending the RIL API call to a proxy at a first interface;
  - (c) at the proxy, converting the RIL API call to a command understood by a radio driver;
  - (d) transmitting the radio driver command from the proxy to the radio driver at a second interface;

- (e) transmitting a radio command from the radio driver to the radio at a third interface; and
- (f) performing the specific action at the radio.

18. (Previously Presented) The method of claim 17 further comprising:

- (g) in response to successfully performing the specific action, sending a success code from the driver to the proxy and from the proxy to the one of the plurality of modules that generated the RIL API.

19. (Original) The method of claim 18 wherein the RIL API, command and success code are associated with an identifier linking them together and linking them to the one of the plurality of modules that generated the RIL API call and wherein the radio driver receives the success code, and, using the identifier, matches the success code with the one of the plurality of modules that generated the RIL API call and sends the success code to the one of the plurality of modules that generated the RIL API call.

20. (Previously Presented) The method of claim 19 further comprising the step of:

- (h) generating a notification at the radio in response to detecting data that needs to be reported to one of the plurality of modules;
- (i) sending the notification to the radio driver.

21. (Previously Presented) The method of claim 20 further comprising the step of:

- (j) sending the notification from the radio driver to the proxy.

22. (Previously Presented) The method of claim 21 further comprising the step of:

- (k) sending the notification from the proxy to at least one of the plurality of modules.

23. (Original) The method of claim 20 wherein the data that needs to be reported comprises an incoming phone call to the radio.
24. (Original) The method of claim 20 wherein the data that needs to be reported comprises a signal strength change in the radio.
25. (Original) The method of claim 20 wherein the one of a plurality of modules is a TSP.
26. (Original) The method of claim 20 wherein the one of a plurality of modules is a SIM manager.
27. (Original) The method of claim 20 wherein the one of a plurality of modules is an emergency application for generating emergency calls.
28. (Original) The method of claim 20 wherein the one of a plurality of modules is a WAP layer.
29. (Original) The method of claim 20 wherein the one of a plurality of modules is a TAPI interface.
30. (Original) The method of claim 20 wherein the one of a plurality of modules is an ExTAPI interface.
31. (Original) The method of claim 20 wherein the one of a plurality of modules is connected to an application program module and receives instructions from the application program module to generate the RIL API call.

32. (Original) The method of claim 31 wherein the instructions provided by the application program module comprise instructions defined by the one of a plurality of modules and wherein the instructions are converted to the RIL API calls by the one of a plurality of modules.